

**SYSTEM AND METHOD FOR TRAFFIC MANAGEMENT CONTROL IN A DATA
TRANSMISSION NETWORK**

Clifford F. Sharp
2553 Via Merano
Del Mar, CA 92014
Citizenship: U.S.A.

Susan Pittman Dark
16007 Chalfont Circle
Dallas TX 75248
Citizenship: U.S.A.

RELATED APPLICATIONS

This application is a Continuation in Part of U.S. Patent Application Serial No. 09/572,112 filed May 17, 2000 and entitled "Intelligent Feedback Loop Process Control System," which is hereby incorporated herein by reference.

TECHNICAL FIELD

This invention relates to data networks control systems and more particularly to a system and method for monitoring and filtering traffic to maintain a constant stream of data flowing in and/or out of a particular location.

BACKGROUND

Data flow in and out of a data source is vital to the proper operation of many enterprises. When this data flow is interrupted, even for the briefest of intervals, a major problem exists. One way to interrupt such data flow is by flooding any communication line (or any network device on the pathway) with so many data packets that the device simply clogs and ceases to function properly. In such a situation, data can be lost, transactions not completed and the flow of commerce halted.

Companies are getting attacked via data flooding by a wide range of flooding mechanisms, including certain types of Denial of Service (DOS) and Distributed Denial of Service (DDOS) attacks that are not specific to an application, but exist in the network.

One known solution to this particular type of problem is manual intervention by a system administrator scrolling on access control list screen to manually select and block an offending IP address.

This, of course, presupposes that the operator even knows which sending IP address(es) is causing the problem. If the rogue sending address keeps changing, the operator (system administrator) is at a loss. The other solutions that we know of are remote intrusion detection sensors that provide manual notification that an attack has been detected and lists the known offending IP address. A third solution is remote monitoring of network conditions. The problem with the solutions to date is the fact that by the time a human can respond to the existing condition, it is already beyond his/her ability to control the traffic and the network is brought down by the intruding traffic overload.

For example, today even assuming a modest speed of, let's say a 1,000 packet/second, about half of the maximum data rate of a T1 channel, the human eye cannot respond that fast to read the IP addresses, digest the information and act before many, many packets enter the system and cause damage. Today a typical enterprise can receive a million hits in an hour, made up of perhaps 4,000,000 or 5,000,000 packets. Such large numbers of packets, when backed up, cause the system to stop functioning.

SUMMARY OF THE INVENTION

The present invention includes a system which implements a three tiered architecture where the database runs on one computer, or across multiple computers, and the kernel and all of its intelligence runs on a separate computer while the applications and management tools run on other computer(s). The system is designed to monitor every tcp/ip packet directed toward a company and to keep track of each packet from each IP address, including all of the bytes of information associated with each packet. The goal of the system and method is to prevent flooding which is defined as a threshold above which data throughput must not go.

If desired, the system can have several threshold's, each dynamically changeable, when a threshold is reached. Threshold, in this context, means that the number of arriving data packets and the accumulated number of bytes of information during a preset time period, (called a primary time slice (PTS)) has met a limit. When this happens certain packets are scanned, captured, and prevented (temporarily) from passing through the system. When the traffic rate falls to the threshold level these captured packets are allowed to then flow through the system. The captured packets are selected based on a comparison of information "sniffed" by the monitor against certain criteria, such as number of packets and the accumulation of bytes per PTS coming from a particular IP address; a known "bad" IP address; and known bad data patterns.

Generally, when it comes to flooding, there are four (or more or less) threshold levels which exist which are percentages of the total maximum band-width of the network. The thresholds can be for example 20, 30, or 40 percent of maximum. If the traffic flow is under the lowest threshold, everything runs smoothly. Once the number of arriving packets and accumulated bytes violate a threshold, the system begins, automatically, the process of choking or holding certain packets. If higher thresholds are violated then more and more

severe action is taken, i.e. more and more packet(s) are prevented from flowing through the system.

One embodiment of the system includes several real time displays, or tools, to analyze, manage, and monitor the data bases, the kernel, and the whole system. One visual display of data and addresses shows the packets that have been choked and, optionally could show why. This information can be displayed locally and can be communicated remotely to any terminal or communication device anywhere in the world. Thus, a user, operating from any terminal with the appropriate user ID and password can make adjustments to the system by changing the different parameters allowing more or less data flow. The operator can, if desired, adjust the criteria for choking data.

The foregoing has outlined rather broadly the features and technical advantages of the present invention in order that the detailed description of the invention that follows may be better understood. Additional features and advantages of the invention will be described hereinafter which form the subject of the claims of the invention. It should be appreciated by those skilled in the art that the conception and specific embodiment disclosed may be readily utilized as a basis for modifying or designing other structures for carrying out the same purposes of the present invention. It should also be realized by those skilled in the art that such equivalent constructions do not depart from the spirit and scope of the invention as set forth in the appended claims. The novel features which are believed to be characteristic of the invention, both as to its organization and method of operation, together with further objects and advantages will be better understood from the following description when considered in connection with the accompanying figures. It is to be expressly understood, however, that each of the figures is provided for the purpose of illustration and description only and is not intended as a definition of the limits of the present invention.

BRIEF DESCRIPTION OF THE DRAWING

For a more complete understanding of the present invention, reference is now made to the following descriptions taken in conjunction with the accompanying drawing, in which:

FIGURE 1 shows the invention working in context to data flowing to/from a network, such as the internet;

5 FIGURE 2 shows in detail the basic blocks which enable the invention;

FIGURE 3 shows a flow diagram of the system operation;

FIGURE 4 shows a seven threads of the system working interactively; and

FIGURES 5-8 show various displays used to monitor system operation.

DETAILED DESCRIPTION

Turning now to FIGURE 1, as packets come from internet 18 (or any source), they travel to data converter 11-1 (11-2) through line 10, which could be a T1, T3, OC48 or any other communication media. Data converter 11-1 can be several data converters which take the data packets off the line and convert that data to an "internal" protocol, such as token 5 ring, FDDI, TCP/IP or other protocol desired by an enterprise. These packets then pass to security system 200-1 one medium 14. When the packets arrive at security system 200-1, each packet will be analyzed, as will be discussed hereinafter. Only those medium packets which make their way through security system 200-1 are delivered to router 12 over 15. Router 12 could be a combination router/gateway or, in fact, router 12 and/or firewall 108 10 could be built directly into security system 200-1 if so desired.

Those packets that manage to make it through security system 200-1 are delivered to enterprise internal (intra) network 13, firewall 108. Connected to intranet 13 can be other 10 intranets, local or remote. For example, intranet 13-N is connected to intranet 13 via medium 17. Connected to any of the intranets can be any type of device such as routers 102, gateways 15 103, servers 104, PCs 105, voice-over IP, protocol systems (VOIP) 106, or any type of devices .01.

The system is designed having a back up line T1-2 so that, if the user desires, information is always delivered from the external network for example from internet 18, over data lines T1 and T2. The packets that come via T1-2 are processed in a similar way as those 20 coming via T1-1, except they do not pass through security system 200-2 until security system 200-2 gets notified that security system 200-1 is not functioning or overloaded. In such an event the information stops flowing from T1-1 and starts flowing from T1-2. This is a back-up system and security systems 200-1 and 200-2 can, for example, communicate with each other over medium 109. Database 19 advantageously is connected to both systems 200-1 and 25 200-2 and is used to store the information analyzing every packet that comes via T1-1 or T1-2.

Applications 107 can be remote from security systems 200-1 and 200-2 if desired and, connected via intranet 13, or they can be local to security system 200-1. The preferred arrangement would be for applications 107 to be in a separate CPU from the one processing system 200-1. They can be run from a location via internet 18 if needed. In that case, the
5 information would come in on modem T1-1 and T1-2 as any other information would come in to the system. This, as will be discussed, allows a user to remotely access the system, monitor the system and make changes to the system as desired.

Data from applications 107 is advantageously binary encrypted so when it's run from anywhere, it is difficult to decrypt. As data flows into security system 200-1 (200-2) the packets are analyzed and do not pass to router/Gateway 12 unless they are logged through the system as will be discussed hereinafter.

FIGURE 2 shows a block diagram, by way of example only, of security system 200: Data enters, via medium 14, to CPU interface 20, which advantageously could be a Sun Ultra Computer running operating system 24 which could be Solaris 8. Administrative interfaces 21 in our example are four visual displays; two of which are data base driven and the other two being visual displays of the kernel in real time. Data base interfaces 22 are, in our example, Oracle 8i version 8.1.7.0.0, and contain a number of tables based on Oracle 8 architecture. Notification system 23 is an alarming system that involves sending out e-mail and pager alerts as packets get analyzed and choked. Filtering processes 25 are Solaris 8 loadable kernel modules that actually filter or disallow packets from passing through the system. SMI 26 is self-monitoring intelligence that is continuously monitoring the complete state of security system 200-1 (200-2). Network interface 27 processes data packets to external interface 15 using Sun Fest Ethernet PCI NIC Internet access. Notification system 23 sends pages, e-mail and/or any other message type, advantageously via the SMTP protocol through external interface 14 using, for example the Sun Solaris Mailx program. Threshold settings control 28 allows for fixed and variable settings.

Commands interface is a Unix shell and/or DOS shell command line interface to the system. Interface allows external issue of commands that change the internal configuration parameters of the system. Along with changing the internal configuration, the configuration database is updated simultaneously and automatically. The following is a list of
5 configuration items that can be altered:

- Status - tells the system to list all the configuration items that can be altered and their current status;
- Email - turn on/off email alerts;
- Pager - turn on/off pager alerts;
- Iptrack - turn on/off writing to the iptrack database;
- Ipchoke - turn on/off writing to ipchoke database;
- Ipicmp - turn on/off writing to ipicomp database;
- Message - turn on/off message, log reporting and change the message reporting level;
- Trace - turn on/off error log reporting and alter the trace level reporting
- Ptslice - modify the PTS period; and
- Threshold s 1-4 - turn on/off each threshold level.

FIGURE3 shows system diagram 300. Packets enter via box 301 via medium 14 (FIGURE2) and each packet is analyzed. Routine 302 looks at the packet to see if the IP address of the packet has sent any previous bad information. Also a tally is maintained in database 19 (FIGURE1) of how many bytes each packet contains and the number of bytes of the packet is added to the total for that same IP address. The system also keeps track of how the packet size fits with respect to the maximum packet size of the other incoming data. Also tallied are the amount of bytes since the last primary time slice (PTS) has been stored along with, the accumulated bytes for the whole time the system has been running for that IP
20
25

address. The date when an IP address first sent a packet through the system, along with the date of the last packet is maintained along with the current number of bytes for that PTS all of which are stored in database 19 via routine 303.

Analysis 304 involves determining if the number of packets and the accumulated
5 bytes per IP address over each PTS, taken as a whole, is enough to violate any of the four
preset threshold levels.

Analysis 304 also determines, on a continuous basis, if the packet byte accumulation
rates warrant the addition or subtraction of IP addresses to the list of chokes as compared to
each threshold level and preset packet rate.

Threshold compare 305, compares the number of bytes during the last PTS where it
10 first sorts the total number of bytes to each IP address by descending number of bytes. In
other words, the highest number of bytes by IP address gets sorted to the top. If the threshold
has been violated box 306, if the total number of bytes that came through for all IP addresses
during that PTS is greater than one (or more) of the threshold levels or if the packet rate
15 violates a preset packet rate threshold. When a level of violation happens, 307 a
determination is made as to which level has been violated. If any threshold is violated, then
certain data packets are choked from flowing through the system.

Once a threshold is violated the system begins to disallow packets from certain IP
addresses to pass through the system. Since the system has already and continuously, keeps
20 track of the total bytes as well as the number of packets for every IP address sorted in
descending order it now uses that tally to choke a certain percentage of traffic. The system
does this by looking at packets from IP addresses that are known to have sent bad packets in
the past. If choking only these packets is not enough to bring the bandwidth down below a
particular threshold level, the system chokes the next IP address on the list having the largest
25 data volume per PTS. If this total accumulation of bytes as well as the packet rate for this IP
address brings total bandwidth down to threshold level then the system stops adding IP

addresses to the choke list. If the throughput still remains over threshold, more and more IP addresses are added to the choke list until the proper throughput is achieved.

During this time the system is simultaneously determining if there is a violation of any other thresholds. If the next higher threshold has been violated, then the system goes through the same process, but looking for a higher percentage level to choke. If the system goes down a threshold, then some of the IP addresses on the choke list are removed. Where the threshold levels are lowered, the cached (choked) addresses are allowed through. Once the database and cache are emptied for that threshold level, then if need be, more addresses are removed from the choke list. Process 309 & 311 control storage of data, while process 310 sends unchoked data through the system.

Another process that happens at every threshold level simultaneously is anytime a move up or down in threshold level is made, a determination is made to see if bytes that are currently being choked are still active. If they are active, then we leave them on the choke list at the next higher level. Active means that the IP address has accumulated bytes and/or packets during the last PTS. If that IP address is inactive for a PTS, meaning it has not accumulated any bytes or packets in the last PTS , then it is removed from the choke list before other IP addresses are added or removed.

Two important parameters of the system are 1) threshold percentages and 2) primary time slice. Everything happens brand new for every time slice, which advantageously is two seconds but could set to any length desired. Then the way the system prioritizes how traffic is choked is by determining how many bytes and packets are accumulated, if any, during the last PTS for each IP address.

The reason for checking to see if an IP address is still active is to prevent an attacker from continuing to change IP addresses, which could occur if “dead” IP addresses were not removed quickly from the choke list.

FIGURE 4 is a description of what is going on during a primary time slice. There are seven processes operating in the processor for every primary time slice and these processes are repeated at 2 second intervals (assuming a 2 second PTS is selected).

The parameters for disc storage for the database storage can be determined on a 5 customizable level, whether a user needs to store data for an extended period of time beyond the PTS expiration of active or inactive packets. The user could determine the amount of time and volume of data that they need to store and for how long and for how many PTS.

Parameters for database storage and back-up will depend upon the amount of 10 bandwidth that is being recorded and will depend upon when the inactive list is available such as by the hour increments or daily or weekly increments to be backed up to maintain database levels for constant writing during any time of flooding traffic.

FIGURE 5 shows Self Monitoring Intelligence (SMI) display 500 shows 12 components (501-512) of the systems states in real-time dynamically. There are four states for every component which can, if desired, be shown in different colors for quicker identification of system status. The colors in parenthesis (optional) are suggested colors.

Down (in red) - the component is down and NOT working - needs HELP.

Init (in blue) - the component is working and has been initialized but is currently NOT active.

Idle (in yellow) - the component is working, initialized, and running but is currently 20 not processing data.

Run (in green) - the component is working, initialized, running, and actively processing data.

FIGURE 6, shows dynamic visual IP Filter monitoring display 600. Table 601 entitled "IP Address Information" is a real-time dynamic list of each and every packet that the 25 system sees on the internet. Table 602 entitled "IP Filter Information" is real-time dynamic list of every single packet that the system chokes.

603 is the total bytes the system has processed starting when the system was initially started.

604 is the total number of packets the system has processed starting when the system was initially started.

5 605 is the total number of bytes the system has choked starting when the system was initially started.

606 is the total number of packets the system has choked starting when the system was initially started.

FIGURE 7, shows visual IP Tracking database management and display 700 which provides more information about each and every IP address that the system maintains. The first two columns of FIGURE 7, I will call them 700a, 700b, are not shown.

700a is the IP address.

700b is the domain name associated with column 700a's IP address up to the current PTS.

701 is the total accumulated bytes that has been processed for IP address 700a.

702 is the total accumulated bytes during the current PTS for IP address 701a.

703 is the total accumulated bytes during the last PTS for IP address 701a.

704 is the accumulated bytes for IP address 701a up to the last PTS.

705 is the date in which the IP address 701a was first logged.

20 706 is the date the IP address 701a was last updated.

801 is a pull-down menu of data which is associated with this system's configurations. There can be many different configurations associated with this system.

FIGURE 8, shows visual management tool 800 for the system configuration database demonstrating a pull down menu of multiple configurations. The system can store as many 25 different configurations in the database as a company would like.

Although the present invention and its advantages have been described in detail, it should be understood that various changes, substitutions and alterations can be made herein without departing from the spirit and scope of the invention as defined by the appended claims. Moreover, the scope of the present application is not intended to be limited to the

5 particular embodiments of the process, machine, manufacture, composition of matter, means, methods and steps described in the specification. As one of ordinary skill in the art will readily appreciate from the disclosure of the present invention, processes, machines, manufacture, compositions of matter, means, methods, or steps, presently existing or later to be developed that perform substantially the same function or achieve substantially the same result as the corresponding embodiments described herein may be utilized according to the present invention. Accordingly, the appended claims are intended to include within their scope such processes, machines, manufacture, compositions of matter, means, methods, or steps.

10